Kotlin

Programming

in Bangla

বিসমিল্লাহির রহ্মানির রহীম।

আসসালামুয়ালাইকুম। শুরুতে কিছু কথা।

কোটলিন একটি সাধারণ উদ্দেশ্য, স্থায়ীভাবে টাইপ এবং ওপেন-সোর্স প্রোগ্রামিং ভাষা। এটি JVM এ রান করে এবং যে কোনও জায়গায় ব্যবহার করা যেতে পারে, আজকে জাভা ব্যবহার করা হয়। এটি অ্যান্ড্রয়েড অ্যাপ্লিকেশান, সার্ভারের পাশের অ্যাপস এবং আরো অনেক কিছুকে বিকশিত করতে ব্যবহার করা যেতে পারে।

গুগলের বাৎসরিক ডেভলপার কনফারেন্স Google I/O তে এবছর ঘোষণা দেয়া হয় যে এখন থেকে কটলিন Android এর অফিসিয়াল প্রোগ্রামিং ল্যাঙ্গুয়েজ। কটলিন হচ্ছে statically typed programming language যা জাভার মতই JVM এ রান করা যায় আর মজার ব্যপার হচ্ছে একই প্রজেক্টে জাভা এবং কটলিন একই সাথে ব্যবহার করা

যায়। ল্যাঙ্গুয়েজ হিসেবে কটলিন খুবই মডার্ন। এটা জেটব্রেইন্স এর প্রোডাকশন তাই ইন্টেলিজে আইডি এর কটলিন প্লাগিন ব্যবহার করে আপনি জাভা কোডকে এক ক্লিকে কটলিনে কনভার্ট করে নিতে পারবেন। যারা Competitive programming করেন তাদের জন্য একটা ভালো খবর হচ্ছে যে জেটব্রেইন্স আগামী ৩ বছরের জন্য ACM ICPC স্পন্সর হিসেবে চুক্তি স্বাক্ষর করেছে। ২০১৮ সালের ওয়ার্ল্ড ফাইনালে কটলিন ব্যবহার করা যাবে। আর কটলিন অনেকটা আপ্যালের সুইফট এর মতই তাই ধরেই নেয়া যায় যে কটলিন দিয়ে Android App ডেভলপ করলে একই সাথে আইওএস অ্যাপ ডেভলপও কিছুটা শিখে যাবেন। তুলনামূলক নতুন প্রোগ্রামিং ল্যাঙ্গুয়েজ হওয়ায় এখনও কটলিন নিয়ে খুব বেশি টিউটোরিয়াল নাই কিন্তু প্রতিদিনই নতুন নতুন টিউটোরিয়াল আসছে। আমার আজকের লেখাটা মূলত পিসিতে কটলিন দিয়ে প্রথম প্রোগ্রামটা রান করার জন্য। জাভার কোড কটলিনে কনভার্ট করার ফিচার এন্ড্রয়েড

স্টুডিওতে যুক্ত আছে। চাইলে ডেভেলপাররা জাভার কোড কটলিনে কনভার্ট করতেই পারেন।

কোটলিনের বৈশিষ্ট্যগুলি

সংক্ষিপ্ত: কোটলিন অতিরিক্ত কোড লেখা কমিয়ে দেয়। এই Kotlin আরো সংক্ষিপ্ত করুন।

নাল নিরাপত্তা: কোটলিন নাল নিরাপত্তা ভাষা। কোটলিন কোড থেকে NullPointerException (নাল রেফারেন্স) বাদ দেওয়ার লক্ষ্য ছিল।

আন্তঃঅর্থনীয়: কোটলিন সহজেই জাভা কোডটি একটি প্রাকৃতিক পদ্ধতিতে কল করেন এবং কোটালিন কোডটি জাভা দ্বারা ব্যবহার করা যায়।

স্মার্ট cast: এটি স্পষ্টভাবে অপরিবর্তনীয় মানগুলি টাইপ করে এবং স্বয়ংক্রিয়ভাবে তার নিরাপদ কাস্টারের মধ্যে সন্নিবেশ করে।

সংকলন সময়: এটি ভাল পারফরম্যান্স এবং দ্রুত সংকলন সময় আছে।

টুল-বান্ধব: কোটলিন প্রোগ্রামগুলি কমান্ড লাইন এবং কোনও জাভা আইডিই এর মাধ্যমে তৈরি হয়।

এক্সটেনশন ফাংশন: কোটলিন এক্সটেনশন ফাংশন এবং এক্সটেনশন প্রোপার্টি সমর্থন করে। যার মানে এটা তাদের কোড স্পর্শ ছাড়া ক্লাসের কার্যকারিতা প্রসারিত করতে সাহায্য করে।

কটলিন সেটআপ ক্রাঃ

কটলিন করার জন্য আপনি ২ ধরনের উপায় করতে পারেন। একটি হল অনলাইনে কম্পাইলার ইউস করা। আরেকটি হল কম্পিউটারে সেটআপ করা।

ক্তলিন ডাউনলোড ক্রুনঃ CLick

*কিভাবে সেটআপ দিবেন তার ভিদিওঃ **Click***

*অনলাইন কম্পাইলার ইউস করতে পারেনঃ **Click***

Kotlin Hello World Program

কটলিনে এবার hello world লিখুনঃ

```
1.  fun main(args: Array<String>){
2.      println("Hello World!")
3.  }
```

এবার দেখে নিন ভিদিওঃ Click

বইটির পরবর্তী ভার্সন খুব তারাতারি দেব ইনশাআল্লাহ। :)র আপনারা চাইলে আমার ইউটিউবে সাবস্ক্রাইব করে রাখতে পারেন। সেখানে আরও কটলিনের ভিডিও টিউটোরিয়াল আপলোড করব। কারো বুঝতে সমস্যা হলে আমাকে জানাবেন, আমি সাহায্য করার চেষ্টা করব।

আমার ইউটিউবঃ Click

কটলিন শেখার প্লে লিস্তঃ Click

চলুন এবার শিখি কিছু সাধারন সহজ কটলিনের সমস্যা ও তার স্মাধানঃ

Kotlin if Expression

```
1.  fun main(args: Array<String>) {
2.      val num1 = 10
3.      val num2 =20
4.      val result = if (num1 > num2) {
5.        "$num1 is greater than $num2"
6.      } else {
7.        "$num1 is smaller than $num2"
8.      }
9.      println(result)
10. }
```

Output:

```
10 is smaller than 20
```

```
1.  fun main(args: Array<String>) {
2.      val num = 10
3.      val result = if (num > 0){
4.          "$num is positive"
5.      }else if(num < 0){
6.          "$num is negative"
7.      }else{
8.          "$num is zero"
9.      }
10.     println(result)
11. }
```

Output:

```
10 is positive
```

Kotlin Nested if Expression

```
1.  fun main(args: Array<String>) {
2.      val num1 = 25
3.      val num2 = 20
4.      val num3 = 30
5.      val result = if (num1 > num2){
6.
7.          val max = if(num1 > num3){
8.              num1
9.          }else{
10.             num3
11.         }
12.         "body of if "+max
13.     }else if(num2 > num3){
14.         "body of else if"+num2
15.     }else{
16.         "body of else "+num3
17.     }
18.     println("$result")
19. }
```

Output:

```
body of if 30
```

Kotlin when Expression

```
1.  fun main(args: Array<String>){
2.      var number = 4
3.      var numberProvided = when(number) {
4.          1 -> "One"
5.          2 -> "Two"
6.          3 -> "Three"
7.          4 -> "Four"
8.          5 -> "Five"
9.          else -> "invalid number"
10.     }
11.     println("You provide $numberProvided")
12. }
```

Output:

```
You provide Four
```

Using when Without Expression

```
fun main(args: Array<String>){

    var number = 4
    when(number) {
        1 -> println("One")
        2 -> println("Two")
        3 -> println("Three")
        4 -> println("Four")
        5 -> println("Five")
        else -> println("invalid number")
    }

}
```

Output:

```
Four
```

Multiple Statement of when Using Braces

```
fun main(args: Array<String>){
    var number = 1
    when(number) {
        1 -> {
            println("Monday")
            println("First day of the week")
        }
        7 -> println("Sunday")
        else -> println("Other days")
    }
}
```

Output:

```
Monday
First day of the week
```

Using when in the range

```
un main(args: Array<String>){
    var number = 7
    when(number) {
        in 1..5 -> println("Input is provided in the range 1 to 5")
        in 6..10 -> println("Input is provided in the range 6 to 10")
        else -> println("none of the above")
    }
}
```

Output:

```
Input is provided in the range 6 to 10
```

Kotlin for Loop & Array:

```
fun main(args : Array<String>) {
    val marks = arrayOf(80,85,60,90,70)
    for(item in marks){
        println(item)
    }
}
```

Output:

```
80
85
60
90
70
```

```
fun main(args : Array<String>) {

```

```
    val marks = arrayOf(80,85,60,90,70)
    for(item in marks.indices)
      println("marks[$item]: "+ marks[item])
}
```

Output:

```
marks[0]: 80
marks[1]: 85
marks[2]: 60
marks[3]: 90
marks[4]: 70
```

Iterate through range

```
fun main(args : Array<String>) {

    print("for (i in 1..5) print(i) = ")
    for (i in 1..5) print(i)
    println()
    print("for (i in 5..1) print(i) = ")
    for (i in 5..1) print(i)           // prints nothing
    println()
    print("for (i in 5 downTo 1) print(i) = ")
    for (i in 5 downTo 1) print(i)
    println()
    print("for (i in 5 downTo 2) print(i) = ")
    for (i in 5 downTo 2) print(i)
    println()
    print("for (i in 1..5 step 2) print(i) = ")
    for (i in 1..5 step 2) print(i)
    println()
    print("for (i in 5 downTo 1 step 2) print(i) = ")
    for (i in 5 downTo 1 step 2) print(i)
```

```
    }
```

Output:

```
for (i in 1..5) print(i) = 12345
for (i in 5..1) print(i) =
for (i in 5 downTo 1) print(i) = 54321
for (i in 5 downTo 2) print(i) = 5432
for (i in 1..5 step 2) print(i) = 135
for (i in 5 downTo 1 step 2) print(i) = 531
```

Kotlin Function:

```
fun main(args: Array<String>){
    var number = 25
    var result = Math.sqrt(number.toDouble())
    print("Square root of $number is $result")
}
```

Output:

```
Square root of 25 is 5.0
```

```
fun main(args: Array<String>){
    sum()
    print("code after sum")
}
fun sum(){
    var num1 =5
    var num2 = 6
    println("sum = "+(num1+num2))
```

```
    }
```

Output:

```
sum = 11
code after sum
```

Kotlin parameterize function example:

```
fun main(args: Array<String>){
   val result = sum(5, 6)
    print(result)
}
fun sum(number1: Int, number2:Int): Int{
    val add = number1+number2
    return add
}
```

Output:

```
11
```

অনেক ধন্যবাদ আপনাকে বইটি পড়ার জন্য।

My Name is KAZI SHAMIM SHAHAREAR ISLAM

My Youtube: Shaharear

Shaharear